



COVER SHEET

This is the author version of article published as:

Mejias, Luis and Saripalli, Srikanth and Campoy, Pascual and Sukhatme, Gaurav (2006) A visual servoing approach for tracking features in urban areas using an autonomous helicopter. In *Proceedings IEEE International Conference on Robotics and Automation 2006*, pages pp. 2503-2508, Orlando, Florida.

Copyright 2006 IEEE

Accessed from <http://eprints.qut.edu.au>

A Visual Servoing Approach for Tracking Features in Urban Areas Using an Autonomous Helicopter

Luis Mejías, Pascual Campoy
Computer Vision Group
Universidad Politécnica de Madrid
Madrid, Spain
{lmejias,campoy}@etsii.upm.es

Srikanth Saripalli, Gaurav S. Sukhatme
Robotic Embedded Systems Laboratory
Center for Robotics and Embedded Systems
Department of Computer Science
University of Southern California, Los Angeles, California, USA
{srik,gaurav}@robotics.usc.edu

Abstract—The use of Unmanned Aerial Vehicles (UAVs) in civilian and domestic applications is highly demanding, requiring a high-level of capability from the vehicles. This work addresses the design and implementation of a vision-based feature tracker for an autonomous helicopter. Using vision in the control loop allows estimating the position and velocity of a set of features with respect to the helicopter. The helicopter is then autonomously guided to track these features (in this case windows in an urban environment) in real time. The results obtained from flight trials in a real world scenario demonstrate that the algorithm for tracking features in an urban environment, used for visual servoing of an autonomous helicopter is reliable and robust.

I. INTRODUCTION

UAVs (unmanned aerial vehicles) have been an active area of research in recent years. Their use in civilian applications for surveillance and inspection continues to grow due in part to the emergence of cheap low-power vision systems. Vision for flight control encompasses a broad range of research areas object detection and tracking, position estimation, navigation, and multivariable non-linear system modeling and control.

An autonomous helicopter is highly suitable for tasks like inspection and surveillance. The inherent ability of the helicopter to fly at low speeds, hover, fly laterally and perform maneuvers in narrow spaces makes it an ideal platform for such tasks. These abilities are particularly useful in urban structured environments where the features that are to be tracked are very well understood. The structured nature of features (e.g. windows) facilitates vision-based state estimation and control.

The approach presented in this paper combines computer vision with low-level control to achieve precise vision based feature tracking for an unmanned model helicopter. The vision-based system described here acts as an overall controller sending navigational commands to the flight controller which is responsible for robust autonomous flight. The result is an algorithm for vision-based tracking of features using an autonomous helicopter in a structured environment.

In the experiments reported here, the helicopter [Figure 1] is initialized at hover at an arbitrary location near a building. A user on the ground selects features (windows on a building). The helicopter is required to align with and track these features in successive frames. We report the methodology underlying this capability and results from successful flight trials.



Fig. 1. Autonomous helicopter during a flight trial

II. RELATED WORK

An early autonomous navigation system for a model-scale helicopter (the Hummingbird) was reported in [1]. The unique feature of this system was the sole use of GPS as the navigation sensor replacing the Inertial Measurement Unit, which is conventionally favored as the primary navigation sensor. The autonomous helicopters reported in [2]–[4] used a combination of vision and GPS for navigation capability. The onboard DSP-based vision processor provided navigation information such as position, velocity and attitude at an acceptable delay (on the order of 10ms), which was combined with GPS and IMU data for accurate attitude and position measurements.

Vision-based robot control has been an active topic of research [5]–[7]. In [8], a vision augmented navigation system is discussed for autonomous helicopter control which uses vision in-the-loop to control a helicopter. A notable vision-based technique used in autonomous helicopter control, is the visual odometer [9], which provides accurate navigational information (position and velocity) which is combined with inertial measurements. In [10] a vision-based solution is given for safe-landing site detection in unstructured terrain where the key problem is for the onboard vision system to detect a suitable place to land, without the aid of a structured landmark

such as a helipad. In [11] a real time vision algorithm for tracking a landing target is presented. Also the vision algorithms are successfully coupled with the helicopter controller to achieve precise landing. An approach where vision is used as sensor for landing in hazardous terrain is described in [12]

Recent work has also included aggressive maneuvering of helicopters [13] and pursuit-evasion games [14]. In [15] a strategy to visually control a small ducted-fan rotorcraft which could draw its power from power lines is described. In [16] some strategies are outlined for visual obstacle avoidance during power line inspections. Work on window tracking for one dimensional visual control of an unmanned autonomous helicopter is reported in [17]. Obstacle avoidance and navigation in urban canyons using optic flow are described in [18]

III. VISUAL TRACKING AND CONTROL

The visual tracking system described here is responsible for tracking features chosen by the user on the ground. This is done by sending reference commands to the flight controller in order to stabilize and navigate the helicopter.



Fig. 2. Window being tracked during flight

The core of the tracking algorithm consists of a Lucas-Kanade tracker [19], [20]. The goal of the Lucas-Kanade algorithm (a Gauss-Newton gradient descent non-linear optimization) is to minimize the sum of squared error between two images, the template and the image warped back onto the coordinate frame of the template.

$$\sum w(x, y) I_x I_y u + \sum w(x, y) I_y^2 v = - \sum w(x, y) I_y I_t \quad (1)$$

$$\sum w(x, y) I_x^2 u + \sum w(x, y) I_x I_y v = - \sum w(x, y) I_x I_t \quad (2)$$

The value $w(x, y)$ is the Gaussian window used for minimizing the sum of squared error, I_x and I_y are the pixel values, u and v are components of optical flow field in x and y coordinates respectively.

Our algorithm tracks four points corresponding to four corners of a window. For this purpose a variation of the algorithm is used as described in [21]. Two of the four points

need to be initialized by the user. First the user picks two corners (opposite corners of the window). The remaining two corners are searched and detected in the neighborhood of the first two. The criteria to find these remaining two points are based on the eigen values, grey level values, corner image location, etc. The central point of these four corners is then used as a basis for the visual reference.

A. Visual References

During visual processing, since the detection and tracking of objects occurs in image space, the natural output from the tracking algorithm is velocity reference in image space. Essentially this represents the set points to the helicopter in terms of velocity in the body-coordinate frame.

1) *Lateral Visual References* : Once the object of interest is located, its location in the image is used to generate the visual references to the flight controller. The diagram that illustrates the lateral visual servoing is shown below (Figure 3). If the camera is located approximately at the center of mass of the helicopter ($X_{hc} = 0$) the angles β and α will coincide. Since the camera is rigidly attached to the helicopter, the task of visual servoing is to make the angles zero that will align the vehicle with the target i.e, the visual servoing algorithm sends commands such that the value of i reaches $\frac{w}{2}$, where w is the image width and i is the horizontal component of the object location in the image.

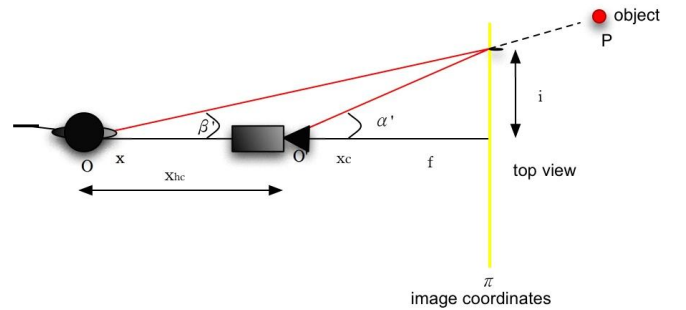


Fig. 3. Helicopter-Camera reference: top view

Based on the Figure 3 i is the value that makes the references higher when the object is far from the center and minimum when is in the image center. The visual signal that commands the helicopter laterally is given by Equation 3.

$$v_{yref} = 2k \frac{(i - \frac{w}{2})}{w} \quad (3)$$

2) *Vertical Visual References*: Using the same approach mentioned above (Figure 4) the task of visual servoing is to drive the value of j to $\frac{h}{2}$, where h is the image height and j is the vertical component of the object location in the image. The visual signal that commands the helicopter vertically is given by Equation 4.

$$v_{zref} = 2k \frac{(j - \frac{h}{2})}{h} \quad (4)$$

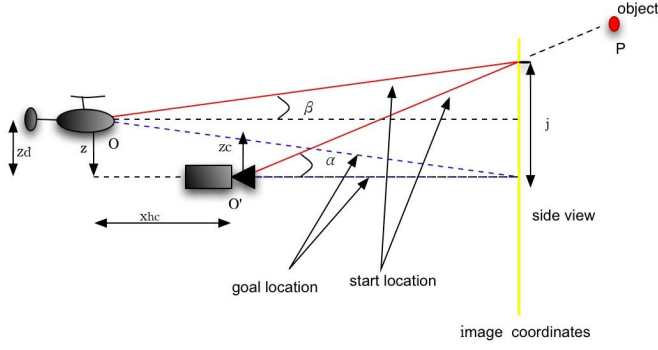


Fig. 4. Helicopter-Camera reference: lateral view

The constant k is used to normalize the visual references to a given value. In the experiments k was chosen to limit the velocity references to ± 2 m/s. Since the references are obtained from the image plane these can be noisy. Before the references are sent to the flight controller a low pass filter is applied. The general form of the filter is as follows

$$X = (1 - k) \times X_{t-1} + k \times X_t \quad (5)$$

where $X = [vy_{ref} \ vz_{ref}]^T$, the vector X_t encodes the current values. The value of k has been empirically chosen to be equal to 0.9.

B. Overall Control Scheme

The overall scheme of the flight controller is shown in Figure 5. The controller is based on a decoupled PID control in which each degree of freedom is controlled separately based on the assumption that the helicopter dynamics are decoupled. The attitude control stabilizes the helicopter in hover maintaining the desired roll, pitch and heading. The velocity and position controllers are responsible for generating the appropriate references to the attitude controller in order to maintain a desired velocity or position, respectively.

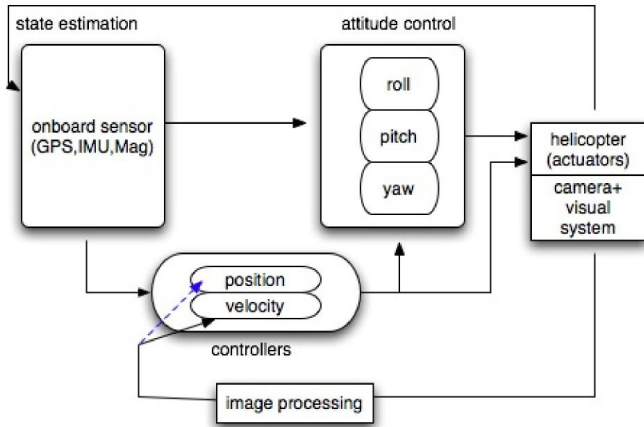


Fig. 5. Flight controller scheme

The attitude controller is implemented as proportional-plus-derivative (PD) control, while the heading controller is implemented with a proportional-plus-integral-plus-derivative (PID) control, the respective output signal are given by

$$\delta_{lat} = K_p(\phi_d - \phi) + K_D \frac{\partial(\phi_d - \phi)}{\partial t} \quad (6)$$

$$\delta_{lon} = K_p(\theta_d - \theta) + K_D \frac{\partial(\theta_d - \theta)}{\partial t} \quad (7)$$

$$\delta_t = K_p(\psi_d - \psi) + K_i \int (\psi_d - \psi) dt + K_D \frac{\partial(\psi_d - \psi)}{\partial t} \quad (8)$$

where: δ_{lat} , δ_{lon} and δ_t are the lateral cyclic, longitudinal cyclic and collective tail commands, respectively. The values K_p , K_i and K_D are the proportional, integral and derivative gains associated with each controller. The position and velocity controller are implemented similarly. Depending on the operation mode the position or velocity controller will generate the collective main rotor command that will displace the helicopter vertically. The values for K_p , K_i and K_D were obtained empirically for autonomous navigation and hover. These gains will be reviewed in the future and tuned appropriately.

IV. THE TESTBED AND THE EXPERIMENTAL TASK

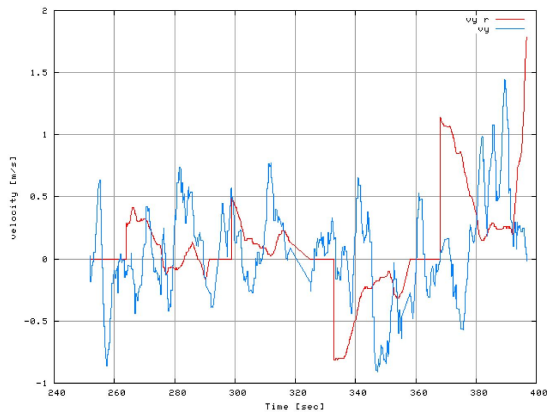
The experimental testbed is based on a gas powered model helicopter twin stroke engine with 52cc and 8 hp, fitted with a Xscale based flight computer augmented with sensors (GPS, IMU, Magnetometer fused with a kalman filter to obtain the helicopter state). For vision processing it has a VIA mini-ITX 1.25GHz computer onboard with 512MB RAM, wireless interface and a color firewire camera for acquiring the images. Both computers run Linux. The ground station is a laptop used to send high-level control commands to the helicopter. is also used for visualization of image data and communications with the onboard image processing algorithm. Communication with the ground station is via wireless Ethernet (802.11b).

V. EXPERIMENTS

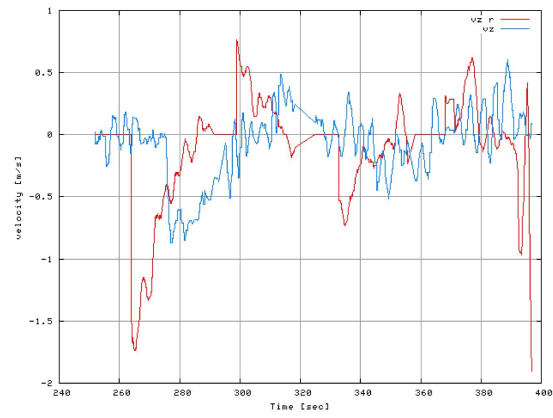
A total of seven experimental trials on two different days were performed to validate the vision system with the flight control. This section presents the results obtained during these trials.

In the experiments the helicopter is commanded to fly autonomously to a given GPS way-point and then the vision algorithm takes control aligning the helicopter to the feature being tracked. These set of flights were performed in an urban area where the features –windows– were the targets of interest to simulate the inspection of a building. From the set of experiments the two most relevant experimental results are presented here. Videos and more information from the flight trials can be downloaded from <http://138.100.76.150/~colibri>. Figures 6 and 7 depict the results for these two trials.

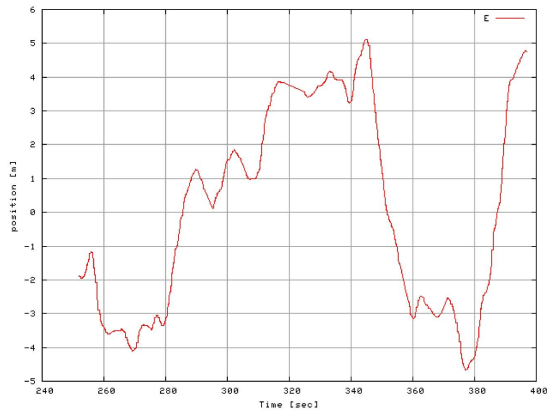
Figures 6(a-e) and Figures 7(a-e) show the location of the helicopter and the references involved while the visual



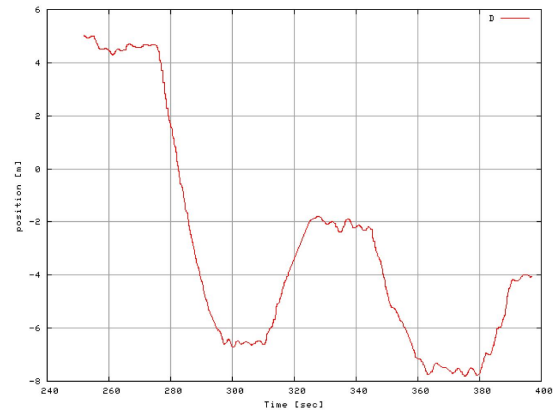
a) v_y and visual references vs time during visual servoing task



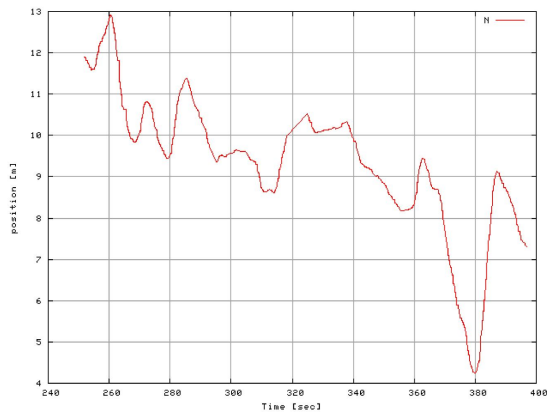
b) v_z and visual references vs time during visual servoing task



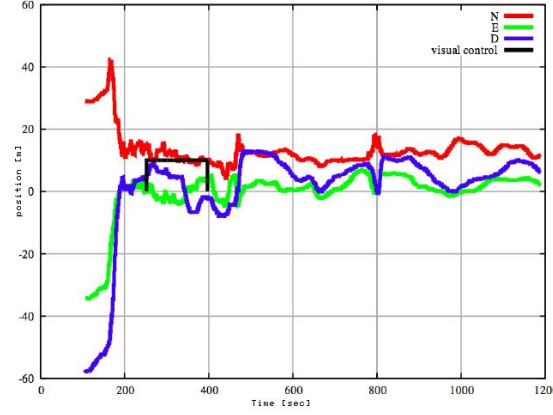
c) Helicopter position (UTM-easting) vs time during visual servoing task



d) Helicopter position (UTM-down) vs time during visual servoing task



e) Helicopter position (UTM-northing) vs time during visual servoing task



f) Helicopter general log for position (UTM coordinates) during the entire flight trial
visual control is high when vision takes control

Fig. 6. Performance of the system during sixth flight trial. Subfigures a) to e) refers to vision based control, subfigure f) shows the helicopter displacements during the entire flight trial

control is performed. Figures 6(a) and 6(b) show the visual references with respect to the helicopter velocity in the lateral and vertical direction. Figure 6(b) correlates the vertical visual references (v_{z_r}) and the helicopter velocity (v_z). This shows the correspondence between two signals that causes the dis-

placements in altitude (down) observed in Figure 6(d). Analogously Figure 6(a) and 6(c), correlates the velocity references (v_{y_r}) with the helicopter velocity (v_y) and lateral displacement (east). Velocity references are given in body frame. For this particular task a negative v_{y_r} causes negative evolution in the

lateral position of the helicopter¹. Negative values vz_r will cause the helicopter to move up (positive evolution in D) and positive to move down. In general the vision-based controller is able to command the requisite velocities that are required for the helicopter to successfully track the features (in our case windows). From the Figures 6(a-b) and Figures 7(a-b) it can be seen that when there is a noticeable change in the targets position in the image plane the algorithm produces the requisite visual reference commands to the velocity controller on the helicopter. This then causes a change in the velocity of the helicopter and subsequently a change in the position of the helicopter such that it is able to align and track the target. Although we do not know the global coordinates of the target, we can see that the reference commands converge to zero which implies that the helicopter has the target in the center of its image coordinate frame.

The behavior observed in the longitudinal velocity in Figures 6(e) and 7(e) is undesirable but understandable since during the task vz_r is fixed to zero which does not guarantee that the helicopter keeps its longitudinal position, since we are controlling velocity not position. A combination of the two approaches, i.e, position and velocity control will cause a better general behavior.

The rate at which the visual processing sends command to the flight controller was one of the most critical parameters. We have noticed that the higher the vision system sends commands to the flight control, the smoother is the resulting helicopter behavior. That is expected since vision sends input reference commands (in term of sample rates) to a high level controller. The low sample rate cause long reaction time and poor perturbation rejection.

In these experiments special emphasis was placed on the speed. In the current version the visual processing rate is 20 fps. The fact that the tracking algorithm only tracks 4 points allows the vision system to run at this frame rate.

VI. CONCLUSION AND FUTURE WORK

We have experimentally demonstrated a robust and consistent approach to visually track urban environment features and navigate towards them using an autonomous helicopter. Such a capability is useful and essential for aerial robots. We performed seven experimental trials on different days to test the effectiveness of the approach. The results presented in this paper show that the visual servoing strategy is effective for tracking windows in urban environments using corners as features. Its believed that the algorithm can be used for other UAV tasks that require visual servoing. The approach was able to track features even with changes of 10° in roll. By requiring a human to select salient features in the environment that should be tracked the computational burden of the algorithm is reduced such that it is possible to run it in real time on an embedded platform on the helicopter. Further in real missions

it is not inconceivable that such a "user interface" would be required for effective use of the helicopter by human operators.

Future work includes an extensive set of experimental trials of the vision approach using a variety of visual features to be tracked and its integration with the flight controller. Additionally we plan to tune the gains of the flight control algorithm for better performance. Also one of the goals is to apply this approach to the inspection of power lines in a real scenario.

ACKNOWLEDGMENT

This work is sponsored by the Spanish Science and Technology Ministry under the project CICYT DPI2004-06624. The authors would like to thank Jorge Leon for support with flight trials.

REFERENCES

- [1] A. R. Conway, "Autonomous control of an unstable helicopter using carrier phase gps only," Ph.D. dissertation, Stanford University, March 1995.
- [2] O. Amidi, "An autonomous vision-guided helicopter," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, 1996.
- [3] R. Miller, B. Mettler, and O. Amidi, "Carnegie mellon university's 1997 international aerial robotics competition entry," in *International Aerial Robotics Competition*, 1997.
- [4] M. Jun, S. I. Roumeliotis, and G. S. Sukhatme, "State estimation of an autonomous flying helicopter," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 1999, pp. 1346–1353.
- [5] Y. Ma, J. Kosecka, and S. S. Sastry, "Vision guided navigation for a nonholonomic mobile robot," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 521–537, June 1999.
- [6] A. Lazanas and J. C. Latombe, "Landmark-based robot navigation," in *proceedings of the Tenth National Conference on Artificial Intelligence*, 1992, pp. 816–822.
- [7] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," in *IEEE Transaction on Robotics and Automation*, vol. 12(5), October 1996, pp. 651–670.
- [8] M. Bosse, "A vision augmented navigation system for an autonomous helicopter," Master's thesis, Boston University, 1997.
- [9] O. Amidi, T. Kanade, and K. Fujita, "A visual odometer for autonomous helicopter flight," in *Proceedings of the Fifth International Conference on Intelligent Autonomous Systems (IAS-5)*, June 1998.
- [10] P. J. Garcia-Padro, G. S. Sukhatme, and J. F. Montgomery, "Towards vision-based safe landing for an autonomous helicopter," *Robotics and Autonomous Systems*, vol. 38, no. 1, pp. 19–29, 2001.
- [11] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Visually-guided landing of an unmanned aerial vehicle," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 371–381, June 2003.
- [12] A. Johnson, J. F. Montgomery, and L. Matthies, "Vision guided landing of an autonomous helicopter in hazardous terrain," in *IEEE International Conference on Robotics and Automation*, May 2005, pp. 3977–3982.
- [13] V. Gavrillets, I. Martinos, B. Mettler, and E. Feron, "Control logic for automated aerobatic flight of miniature helicopter," in *AIAA Guidance, Navigation and Control Conference*, Monterey, CA, USA, Aug 2002.
- [14] R. Vidal, O. Shakernia, H. J. Kim, D. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: Theory, implementation and experimental evaluation," *IEEE Transactions on Robotics and Automation*, pp. 100–107, Oct 2002.
- [15] I. Golightly and D. Jones, "Visual control of an unmanned aerial vehicle for power line inspection," in *Proceedings 12th International Conference on Advanced Robotics*, July 2005, pp. 288–295.
- [16] M. Williams, D. I. Jones, and G. K. Earp, "Obstacle avoidance during aerial inspection of power lines," *Aircraft Engineering and Aerospace Technology*, vol. 73(5), pp. 472–479, 2001.
- [17] L. Mejias, S. Saripalli, G. Sukhatme, and P. Campoy, "Detection and tracking of external features in a urban environment using an autonomous helicopter," in *IEEE International Conference on Robotics and Automation*, May 2005, pp. 3983–3988.

¹Note that the lateral and longitudinal positions correspond to easting and northing in these particular set of experiments. This is because the helicopter was flying with its nose pointing towards north. This is not true in all cases.

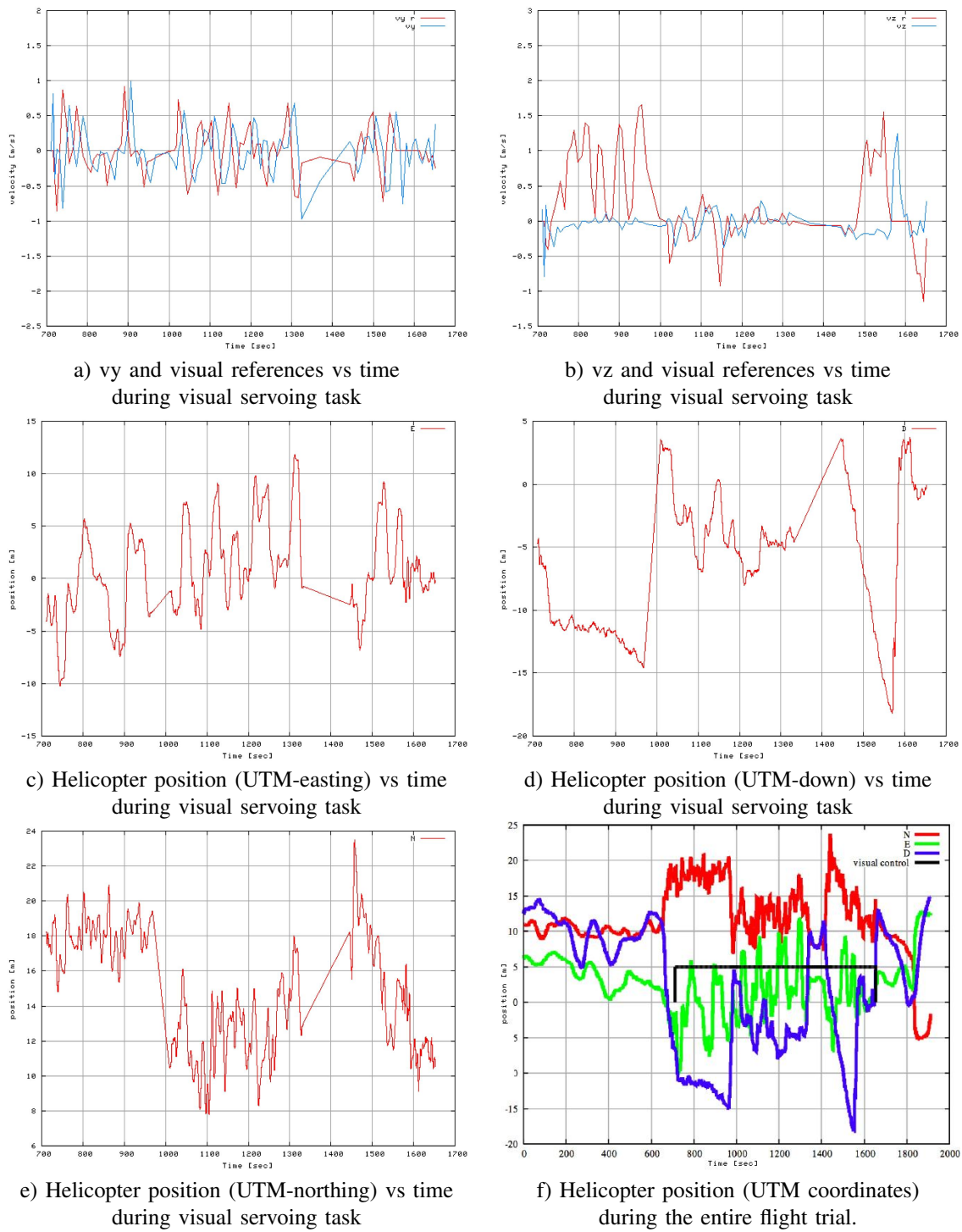


Fig. 7. Performance of the system during seventh flight trial. Subfigures a) to e) refers to vision based control, subfigure f) shows the helicopter displacements during the entire flight trial

- [18] S. Hrabar and G. S. Sukhatme, "Omnidirectional vision for an autonomous helicopter," in *IEEE International Conference on Robotics and Automation*, Taiwan, May 2003, pp. 2799–2804.
- [19] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. of the 7th IJCAI*, Vancouver, Canada, 1981, pp. 674–679.
- [20] C. Tomasi and T. Kanade, "Shape and motion from image streams - a factorization method," *Proc. of the National Academy of Sciences of the United States of America*, vol. 90, no. 21, pp. 9795–9802, November 1993.
- [21] Bouguet Jean Yves, "Pyramidal implementation of the lucas-kanade feature tracker," Microsoft Research Labs, Tech. Rep., 1999.